

FASTPROP ENVIRONMENTAL EFFECTS FOR CONSTRUCTIVE SIMULATIONS

Chuck Lamar
U.S. Army Space And Strategic Defense Command

Wendell Cook
Welman Gebhart
Nichols Research Corporation

ABSTRACT

FastProp is an ultra-fast propagation simulation that models atmospheric and weather environmental effects on propagation of RF and optical signals for constructive simulations. It has been developed under the management of the US Army Space and Strategic Defense Command for the Defense Modeling and Simulation Office (DMSO). FastProp has been successfully interfaced with the EADSIM (Extended Air Defense Simulation) constructive simulation during runtime execution, and is currently being upgraded to interface with the EADTB. Since constructive simulations request propagation data for a variety of sensors and sensor/target geometries there are multi-resolution requirements on generating weather scene objects and in computing the propagation effects at different sensor wavelengths and fields of view. To minimize storage and computation times associated with generating weather scenes over a large area at very high resolutions, FastProp uses a methodology that includes a hierarchical weather object data base, stochastic indexing, and a line of sight (LOS) intersection algorithm. The stochastic indexing allows the repeatable construction of weather scenes from the hierarchical weather object database without requiring large 3-D or 4-D gridded arrays of weather scene data. The LOS intersection algorithm provides an efficient technique for determining the weather objects that affect the signal propagation at the appropriate level of resolution. The FastProp methodology and its critical components will be described in this briefing, and example interface requirements with the EASIM and EADTB constructive simulations will be discussed.

1. INTRODUCTION

U.S. Army Space and Strategic Defense Command (USASSDC) has long developed and used simulations to support the acquisition process. The Extended Air Defense Simulation (EADSIM) has provided a force-on-force modeling capability to support this process. However, the crucial element of weather has to date been modeled inadequately if at all, or has required extensive computation resources. A typical theater missile/air defense scenario may have 10s to 100s of platforms. Considering the effects of weather on each valid line-of-sight among all the platforms for scenario intervals on the order of a second or less can quickly bring an otherwise fast running

air defense simulation to a crawl. Most often this will not allow an acceptable time-frame for performing studies required to support acquisition.

FastProp works towards overcoming this problem by using a distributed simulation approach that isolates the computations required for weather effects from the calculations performed in EADSIM (flight processing, C31, sensors, etc.). FastProp delivers to EADSIM propagation effects based on a request/response protocol data unit (PDU) scheme. These are experimental PDU's whose structure are consistent with others used for Distributed Interactive Simulation (DIS). These effects are then used by EADSIM to perform signal to noise (SNR) calculations and determine if detection has occurred. The result is an efficient methodology which allows the systems analyst to assess systems effectiveness over a wide range of weather conditions. This is a key requirement in any serious weapon/sensor acquisition exercise.

The FastProp team consists of the following:

USASSDC and NRL. The agencies have provided management to this effort which is funded by DMSO. USASSDC being the prime managing agency. NRL has managed the MIT/LL contribution to FastProp.

Hughes STX. Hughes STX developed the methodology for describing the weather using hierarchical objects (HeFeS - Hierarchical Feature Simulation). These objects can be generated quickly and stored efficiently using stochastic indexing. The inputs to this process are generated by PARGET; a software module which is driven by climatological databases.

Nichols Research Coloration. NRC has provided the FastProp preprocessor which includes a graphical user interface (GUI) for defining weather scenes and sensor types, execution of standard physics models to develop tables used by the runtime code. Integrated within the preprocessor is PARGET and FastView. NRC has also developed a software module used by the runtime code which calculates the LOS intersections with the weather objects and weather effects using the look-up tables developed in the preprocessor. In addition, a capability has been developed to generate GRIB files to be used off-line by the Extended Air Defense Testbed.

Decision Science Associates. DSA has developed the software required on both the server side (FastProp) and the client side (EADSIM) to support the network interface. This interface includes interface buffers which store and process incoming/outgoing PDU's between the two simulations. The communications itself is done using UNIX sockets configured in a user datum protocol mode (UDP). On the FastProp side, DSA has built the driver which makes the call to the NRC supplied weather effects module.

Georgia Tech Research Institute / MIT Lincoln Laboratory. These two team members have provided consultation on RF propagation and optical propagation respectively.

They have been involved in the recommendation of physics models to be used in the preprocessor, and validation of the rapid LOS techniques used to calculate weather effects.

Teledyne Brown Engineering. TBE is the primary developer/integrator of the Extended Air Defense Simulation (EADSIM). They have integrated code developed by DSA (the interface buffer) and NRC/Hughes STX (FastView) in order to support the use of the FastProp server for providing the particular weather effects used by the EADSIM sensor models. Enhancements have also been made to the EADSIM GUI to support the user in setting up sensors and scenarios for use with FastProp.

2. REQUIREMENTS

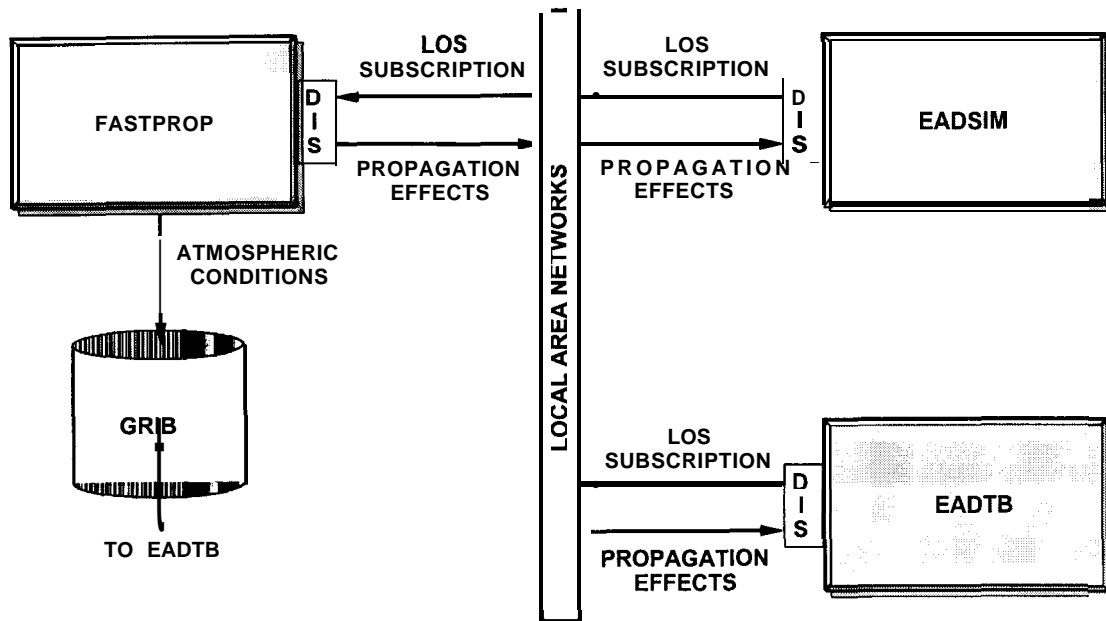
The primary requirement FastProp is to rapidly provide the user with weather effects on sensors and weapons across the visible, IR, and RF spectrums. It should also be easy to use for the analyst who does not have any meteorological training other than a familiarity with the standard terms used to describe weather by weather observers.

The outputs provided to EADSIM in real time are the transmission and background radiation for optical systems; transmission and backscatter for RF systems. Recently, a desire to supply the EADTB with a grid of physical parameters was added as an additional requirement. A file meeting the meteorological community's GRIB standard can now be generated by FastProp during preprocessing which includes ground level temperature, pressure, relative humidity, and velocity as well as the base height, top height, and liquid water content of clouds. At present there is no real-time communication scheme for interfacing with EADTB.

3. METHODOLOGY

The weather modeled in FastProp is broken out into two primary parts: 1) a layered atmosphere, which is characterized by aerosols, temperature, pressure, and humidity, and 2) clouds/rain which are characterized by their location, size and shape, liquid water content, and rain rate. The layered atmosphere can be specified by the user or it can be approximated by the preprocessor based on the location and time of year specified by the user. The cloud scenes are generated by PARGET. They are then placed in the scene in such a way to satisfy statistical data related to the location and time.

During an EADSIM scenario, a request for the weather effects along a path between an observer and target is made to FastProp. FastProp rapidly calculates the effects and returns them to EADSIM. With this distributed approach, EADSIM does not have to devote any computational resources to managing a weather scene or calculating the effects. This architecture is shown in Figure 1.



• TO BE HLA COMPATIBLE

Figure 1 Near Term Architecture

As shown in Figure 2, FastProp is made up of two primary parts; the preprocessor and the runtime code. The preprocessor includes the graphical user interface and performs all the functions of generating the cloud scene given the user inputs, and preparing look-up tables required by the runtime code. The runtime code reads these files during initiation and interacts with the interface buffer to process the requests which come across the network from EADSIM. The postprocessor allows the user to review key parameters of the run such as the number of requests that were made, and how many cloud intersections were logged.

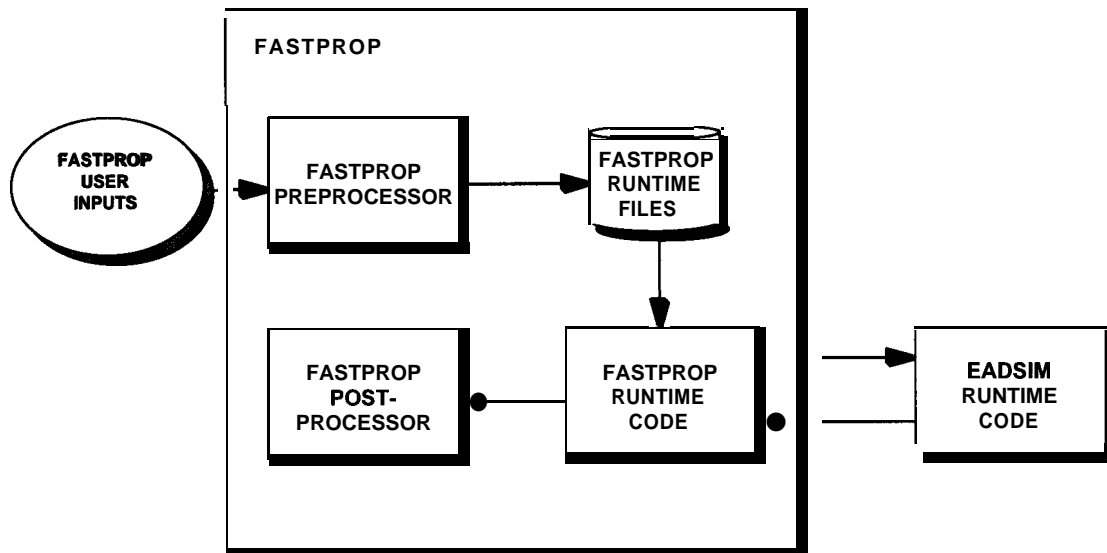


Figure2 FastProp Elements

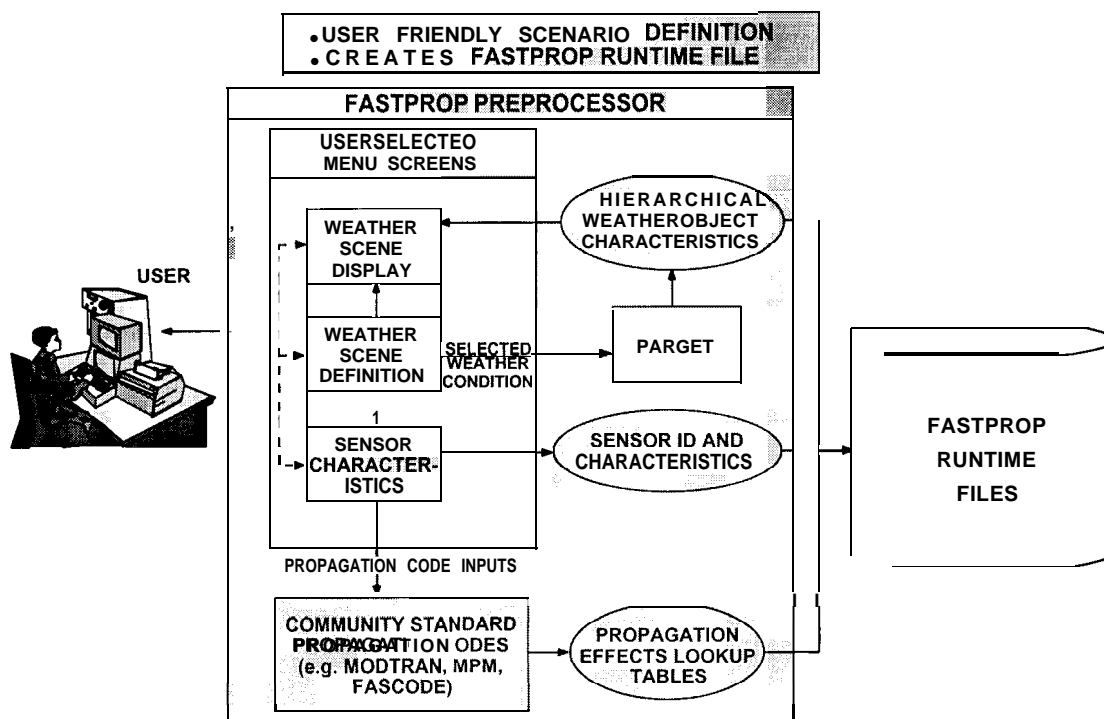


Figure 3 FastProp Preprocessor

The FastProp preprocessor is highlighted in Figure3. The user interfaces with the preprocessor through a GUI which allows him or her to define the weather scenario and define sensors which will be used in the scenario. The weather scenario and sensor definitions are saved to ASCII files. After the weather inputs and sensors have been defined, a click of a button will cause the preprocessor to open the weather and sensor files and set up data structures in a format required for use by the physics codes. These codes are community standard codes which include MODTRAN (for visible and IR) and MPM (for RF) and FASCODE for use with laser systems. The

physics codes then generate the propagation effects lookup tables for the layered atmosphere which are used during runtime. Also, the data in the weather files are used by PARGET in order to generate the cloud objects. These objects are then stored for use by the runtime code.

The GUI which is presently being finalized will give the user the ability to define parameters consistent with the language used by weather observers (air crews, air traffic controllers, etc.) The observable will include cloud ceiling, base height, coverage conditions, humidity, temperature, and precipitation. The user will also be able to designate the location and time of year. These parameters will be used to drive the PARGET module. The top-level weather specification window is shown in

WEATHER EDIT	
<input type="checkbox"/> POINT <input type="checkbox"/> AREA <div style="display: flex; justify-content: space-between;"> <div>SW NE</div> <div> <div style="display: flex; justify-content: space-between;"> <div>LAT</div> <div>LONG</div> </div> <div> <input type="text"/> <input type="text"/> <input type="text"/> </div> </div> </div> <div>REGION <input type="text" value="NORTHEAST ASIA"/></div>	<div>PHENOMENOLOGY</div> <div> <input checked="" type="checkbox"/> CLOUDS <input type="checkbox"/> SNOW <input type="checkbox"/> VISIBILITY <input type="checkbox"/> RAIN <input type="checkbox"/> FOG <input type="checkbox"/> AMBIENT ATMOSPHERE </div>
<div> <input type="checkbox"/> SPECIFIC <input type="checkbox"/> SEASON <input type="checkbox"/> USE YEARLY AVG. </div> <div> <div style="display: flex; justify-content: space-between;"> <div>MM</div> <div>DD</div> <div>YYYY</div> </div> <div> <input type="text"/> <input type="text"/> <input type="text"/> </div> </div> <div> <div>LOCAL HOUR (I-24)</div> <input type="checkbox"/> </div>	
<div>CASE: <input type="text" value="OVERCAST"/></div> <div>VALUE: <input type="text" value="100"/></div> <div> <input type="checkbox"/> BEST <input type="checkbox"/> WORST <input type="checkbox"/> AVERAGE </div> <div> <input type="checkbox"/> PERCENTILE </div>	

Figure 4 User Interface

Figure 4.

The runtime portion of FastProp can be run in an online and an off-line mode. The off-line mode is used to generate GRIB file inputs for EADTB and to support verification of FastProp. The online version interacts with the DSA supplied code which handles the buffering of requests and responses and the socket interface.

When running in the online mode with EADSIM, FastProp carries out calculations which derive the weather effects. The systems models which characterize the sensor and weapon performance reside in EADSIM. For the present off-line implementation with EADTB, the systems models and the effects models reside in EADTB. FastProp only provides physical attributes of the atmosphere (temperature, pressure, humidity, etc.). This demonstrates that FastProp can support clients with a wide spectrum of needs and requirements. The runtime configuration is shown in Figure 5.

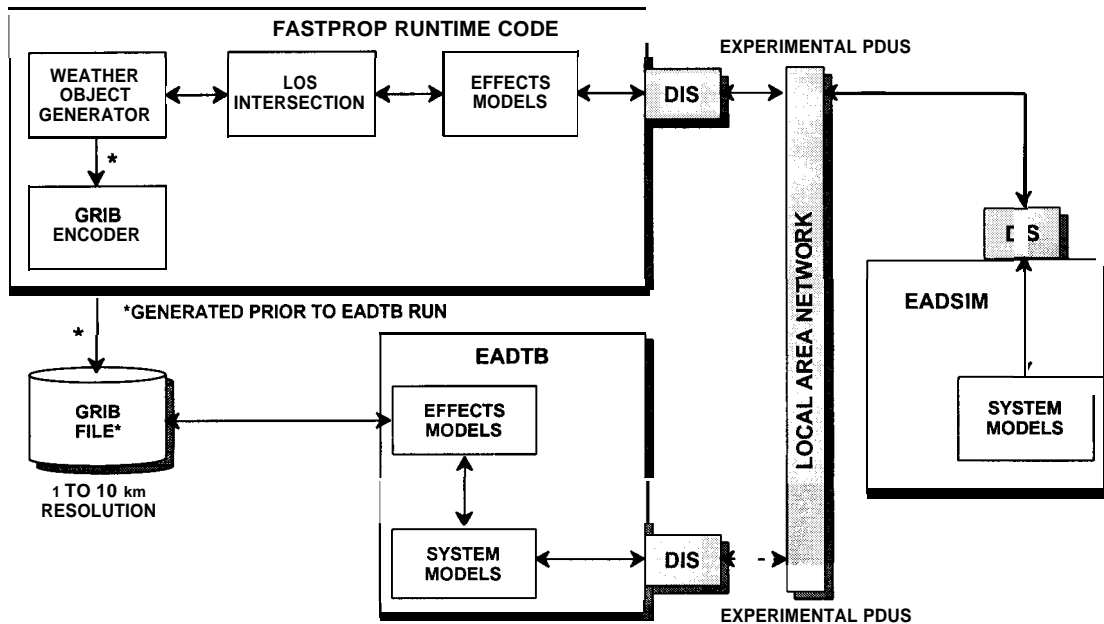


Figure 5 FastProp Runtime Elements

The approach for calculating the effects rapidly is to minimize the integration steps during runtime. The integration over the layered atmosphere is performed over the entire elevation space for different altitudes in the preprocessor and stored in a lookup table. During runtime the LOS algorithm is exercised which determines the weather objects (clouds, precipitation shafts) that are intersected. The extinction-depth product of these objects are then summed with the extinction-depth products from the layered atmosphere to get the total transmission along the path.

The key for the rapid computations is the LOS intersection algorithm and the HeFeS object generation. The object hierarchy is presently comprised of 4 levels: the slab which is typically at the mesoscale level and is made up of rows. The rows are then made up of clouds with scales on the order of 100s of meters. Each cloud can be resolved into meter scale puffs which is the lowest level. There is theoretically no limit to the resolution the process will support, but there is presently no requirement to resolve the objects beyond the puff level. In a typical scenario there may be up to 50,000 puff objects or more in the region, whereas the LOS may actually intersect only 10 of these objects. The majority of these objects can be ignored by using a hierarchical filter process. As shown in Figure 6, the first check is done to determine intersection with the slabs. Because the slabs have a well defined extent in 3D space, the verification of intersection at the slab level is straight forward. The next elements that make up the slab-- the rows, clouds and puffs are generated using the HeFeS methodology that allows them to be specified by a functional relationship. When evaluating a LOS, the cloud scene is rapidly generated internally. Given the intersection with the slab, the scene is only generated for that portion of the slab that could be intersected given the geometry. The level at which this process continues (to

the cloud or puff level) depends on the FOV of the observing system and the range. Close in targets require the algorithm to evaluate intersection at the puff level in the area be evaluated. Longer range targets will cause the algorithm to stop evaluation at the cloud or even the row level. The LOS intersection process takes advantage of this functional relationship to quickly determine which objects are intersected and the thickness of the traversal through the object.

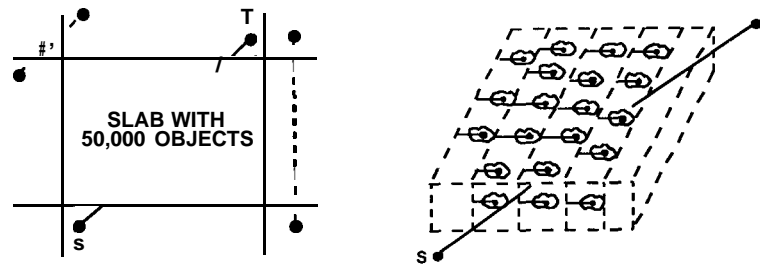


Figure 6 LOS Intersection Methodology

4. REPRESENTATIVE RESULTS

Figure 7 shows a representative weather scene over Fort Hunter Liggett. This scene consists of one slab with dimensions approximately 61 x 64 x 3 km thick. It is comprised of 31 rows, 1300 clouds with 500 meters average horizontal radius and 500 to 3000 meter vertical towers, and 85000 puffs with average radii of 100 meters. This particular view is from EADSIM.

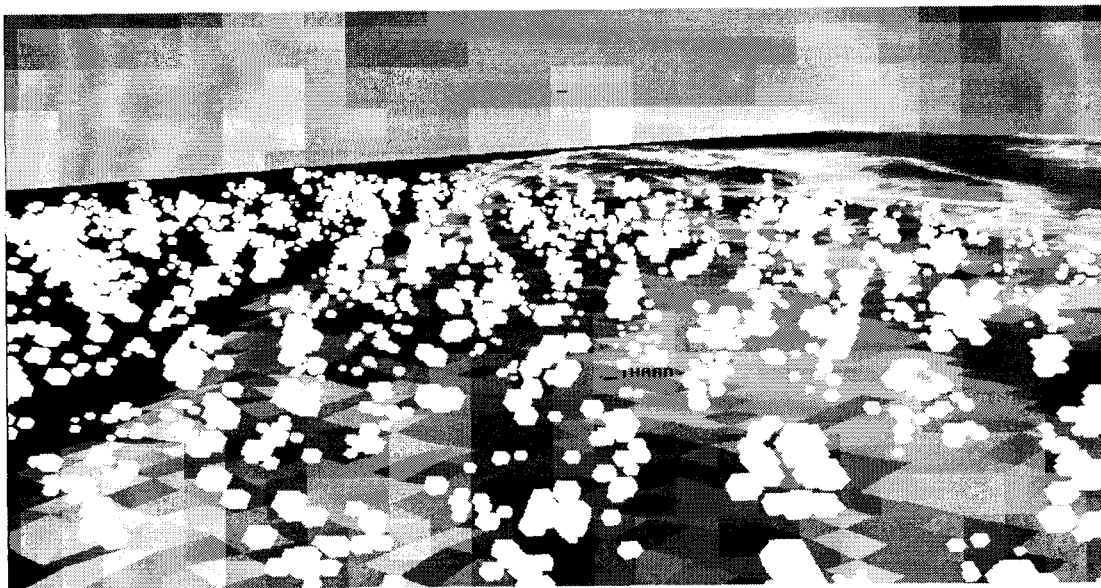


Figure 7 Single slab generated with 22 input parameters

For testing purposes a similar scene was produced and two flight paths for aircraft were defined to fly orthogonally to each other (see Figure 8). The aircraft were set at

2000 meter altitudes. One was aligned approximately along one of the slab rows (Flight Path 1) and the other flying across the rows (Flight Path 2). For this particular scene the cloud ceiling was approximately 1500 meters. A ground observer was placed at the cross-over points of these two flight paths. The aircraft were flown and the intersections and transmissions were calculated for both an LWIR sensor and a 10GHZ radar located at the ground observer point. Figure 9 shows the number of puffs intersected by the LOS paths to the targets on Flight Paths 1 and 2 for two sensors (an LWIR and radar sensors) located at the ground observer point. The radar has approximately 100 times the bearnwidth of the LWIR sensor pixel, and therefore the radar intersects significantly more puffs than the LWIR sensor. However since the cloud extinction of the LWIR signal is much greater than the radar extinction, the radar transmission is through the clouds is much higher than the LWIR transmission, as illustrated in Figure 9 for the Flight Paths 1 and 2, respectively.

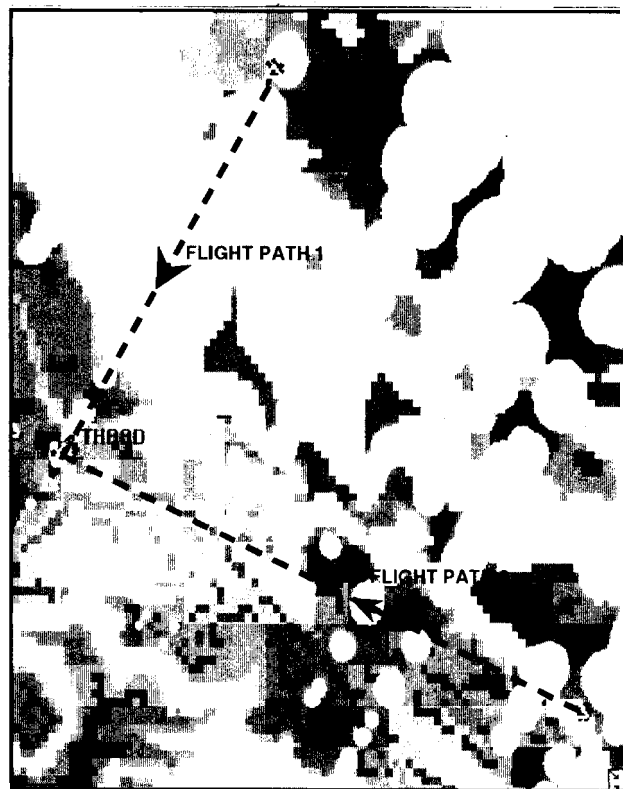


Figure 8 Target Flight Paths

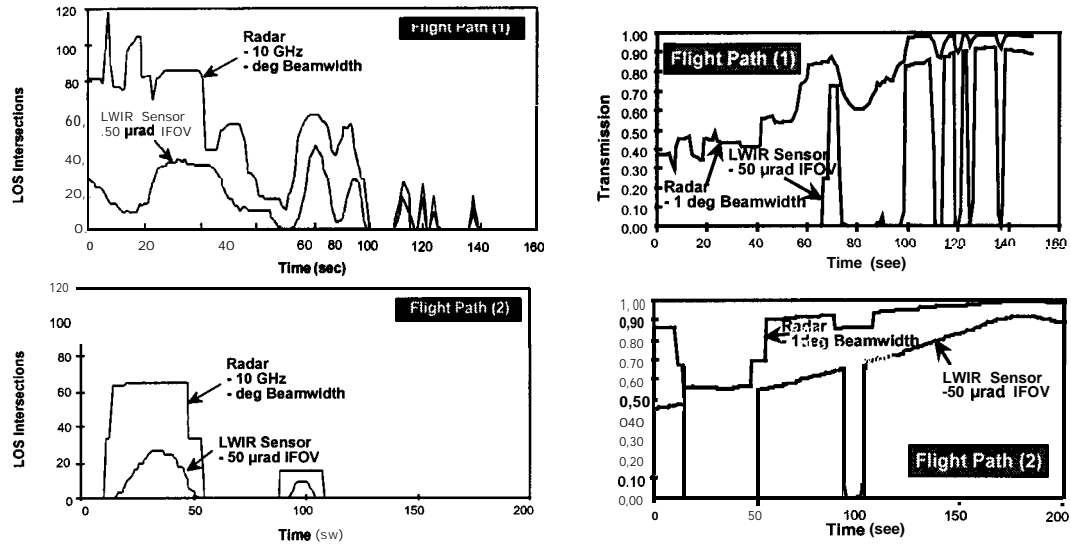


Figure 9 Runtime Results

5. SUMMARY

FastProp provides atmosphere and weather propagation effects for constructive simulations. To date it has been integrated successfully with both the EADSIM (online) and EADTB (off-line using GRIB encoded data). It is compatible with DIS technology, utilizing experimental protocol data units. It has been written in C and is organized in such a fashion as to maximize portability of the preprocessor across platforms. Future enhancements will provide a network interface with the EADTB, the ability to use encoded GRIB data as an input data source, and the incorporation of a 'subscription' protocol in working towards compatibility with the new High Level Architecture (HLA) standard.